A Framework for Intrusion Detection System Using Random Forests and Support Vector Machine

Aye Mon Yi
University of Computer Studies, Mandalay
ayemonyi@gmail.com

Abstract

Due to continuous growth of the Internet technology, it needs to establish security mechanism. However, many current intrusion detection systems (IDSs) are rule-based systems, which have limitations to detect novel intrusions. Moreover, encoding rules is time-consuming and highly depends on the knowledge of known intrusions. Therefore, we propose new systematic framework that apply a data mining algorithm called random forests (RF) and Support Vector Machine (SVM). This system uses Random Forests (RF) for feature selection and parameter optimization and Support Vector Machine (SVM) for intrusion detection. RF provides the variable importance by numeric values so that the irrelevant features can be eliminated. Support Vector Machines (SVM) as a classical pattern recognition tool have been widely used for intrusion detection. First, RF is utilized to preprocess the data and select the most important features to eliminate the insignificant features and optimize parameters. Second, SVM model is used to learn and detect intrusion using selected important features.

Keywords: Intrusion Detection Systems, Random Forests, Support Vector Machine, Feature Selection, Parameter Optimization

1. Introduction

The field of computer security has become a very important issue for computer systems with the rapid growth of computer network and other transaction systems over the Internet. This led to an increase in cyber attacks which necessitates the need for an effective intrusion detection system. Intrusion Detection Systems (IDSs) play a vital role in network security. Network Intrusion Detection Systems (NIDSs) detect attacks by observing various network activities, while Host-based Intrusion Detection Systems (HIDSs) detect intrusions in an individual host.

There are two major intrusion detection techniques: misuse detection and anomaly detection. Misuse detection discovers attacks based on the patterns extracted from known intrusions. Anomaly detection identifies attacks based on the significant deviations from the established profiles of normal activities. Misuse detection has low false positive rate, but cannot detect novel attacks. Anomaly detection can detect unknown attacks, but has high false positive rate.

The proposed approach will be evaluated using the KDD'99 datasets, which were used for the third International Knowledge Discovery and Data Mining Tools Competition [5]. The contest involved building a classifier for detecting computer network intrusions from a very large database of network traffic. Our experimental result is not available yet but the system will show that the detection performance will improve by our approach of using random forests and SVM algorithm.

This paper organized as follows. Section 2 presents the related work that includes machine learning algorithm such as random forests, support vector machine, decision tree. Section 3

describes random forest. Section 4 presents Support Vector Classifier. Section 5 presents about how to implement the system. Section 6 describes about intrusion detection experiments. Section 7 discusses conclusion.

2. Related Work

Lee et al. [13] have proposed a hybrid approach for real-time network intrusion detection systems. They used Random Forest (RF) for feature selection and Minimax probability Machine (MPM) for intrusion detection. They compared their result with previous approaches such as Random Forests, Genetic Algorithm (GA) and Support Vector Machine (SVM), and Filter and SVM. Their experimental results show their approach is faster and more lightweight than the previous approaches.

S. A. Mulay et al. [12] proposed the decision tree based algorithm to construct multiclass intrusion detection system. The classification applications can solve multi-class problems. Decision-tree-based support vector machine which combined support vector machines and decision tree can be an effective way for solving multi-class problems. This method can decrease the training and testing time, increasing the efficiency of the system. The integration of Decision tree model and SVM model gave better results than the individual models. The final results were not available yet, but they believed their intrusion detection system could be faster than other methods.

R Chen et al. [10] used RST (Rough Set Theory) and SVM (Support Vector Machine) to detect intrusions. First, RST was used to preprocess the data and reduce the dimensions. Next, the features were selected by RST will be sent to SVM model to learn and test respectively. The method is effective to decrease the space density of data. The experiments would compare the results with different methods and showed RST and SVM schema could improve the false positive rate and accuracy. Their framework RST-SVM method result had a higher accuracy as compared to either full feature or entropy. The experiment demonstrates that RST-SVM yielded a better accuracy.

3. Random Forests

The random forests [6] are an ensemble of unpruned classification or regression trees. In general, random forest generates many classification trees and a tree classification algorithm is used to construct a tree with different bootstrap sample from original data using a tree classification algorithm. After the forest is formed, a new object that needs to be classified is put down each of the tree in the forest for classification.

Each tree gives a vote about the class of the object. The forest chooses the class with the most votes. RF algorithm is given below [1]:

1. Build bootstrapped sample B*i* from the original dataset D, where |B*i*| = |D| and examples are chosen at random with replacement from D.
2. Construct a tree τ_i, using B*i* as the training dataset using the standard decision tree algorithm with the following modifications:
   a. At each node in the tree, restrict the set of candidate attributes to a randomly selected subset (*x1, x2, x3, … , xk*), where *k = no. of features*.
   b. Do not prune the tree.
3. Repeat steps (1) and (2) for *i = 1, … , no. of trees*, creating a forest of trees τ_i , derived from different bootstrap samples.
4. When classifying an example x, aggregate the decisions (votes) over all trees τ_i in the forest.

If $\tau_i(x)$ is the class of x as determined by tree τ_i, then the predicted class of x is the class that occurs most often in the ensemble, i.e. the class with the majority votes.

Random Forest has been applied in various domains such as modelling [3] [8], prediction [2] and intrusion detection system [4] [13]. Zhang and Zulkernine [4] implemented RF in their hybrid IDS to detect known intrusion. They used the outlier detection provided by RF to detect unknown intrusion. Its ability to produce low classification error and to provide feature ranking has attracted Lee et al. [13] to use the technique to develop lightweight IDS, which focused on single attack.

In random forests, there is no need for cross validation or a test set to get an unbiased estimate of the test error. Since each tree is constructed using the bootstrap sample, approximately one-third of the cases are left out of the bootstrap samples and not used in training. These cases are called out of bag (oob) cases. These oob cases are used to get a run-time unbiased estimate of the classification error as trees are added to the forest.

3.1. Variable Selection with Random Forests

Random Forests were introduced by Breiman for feature (variable) selection and improved predictions for decision tree models. RF consists of several hundred models with randomly selected variable subsets. Random Forests were used with decision tree models by aggregating many tree predictors to obtain an improved predictor. The main idea is that after generating a vast number of trees, they vote for the most popular variables based on performance. Bagging is used in tandem with RF variable selection in order to reduce the variance. In this paper this random forests idea is used to estimate the importance of variables.

RF produces variable importance by numeric values. Feature ranking is performed according to the important value of features. Thus, the variable importance is able to make one easily figure out which features are important or not. We can rank the whole features in descending order with respect to their feature important value, and eliminate the irrelevant feature which is the lowest ranked. In other words, we can select important features. This enables our approach to reduce computational overheads of dataset as well as to enhance the detection rates.

3.2. Optimization for Random Forests

The error rate of a forest depends on the correlation between any two trees and the strength of each tree in the forest. Increasing the correlation increases the error rate of the forest. The strength of a tree relies on the error rate of the tree. Increasing the strength decreases the error rate of the forest. When the forest is growing, random features are selected at random out of the all features in the training data. The best split on these random features is used to split the node. The number of random features (*Mtry*) is held constant. Reducing (Increasing) *Mtry* reduces (increases) both the correlation and the strength. The number of features employed in splitting each node for each tree is the primary tuning parameter (*Mtry*).

To improve the performance of random forests, this parameter should be optimized. RF has these only two parameters: the number of variables in the random subset at each node (*Mtry*) and the number of trees in the forest (*Ntree*), and RF is usually not very sensitive to their values. However, it is important to optimize those two parameters to maximize the classification accuracy. In this paper, we use the

variable importance for the optimal feature selection phase and optimize *mtry* and *ntree* in parameters optimization phase.

4. Support Vector Machine

Recently, support vector machines (SVMs) have been a promising tool for data classification. Its basic idea is to map data into a high dimensional space and find a separating hyperplane with the maximal margin. Consider the problem of separating the set of training vectors belonging to two separate classes,

$T = \{(x_1 y_1),...,(x_l, y_l)\} \in R_l \times \{\pm 1\}$, $x_i \in R^n$,

$y_i \in \{1,-1\}, i=1,\ldots,l$ (1)

Where x_i is a feature vector, y_i is the class of x_i. With a hyperplane

$$\omega.x + b = 0 \quad (2)$$

It makes the positive input and negative input located in opposite sides of this hyperplane. In other words, there exists a pair of argument （ω, b）, make the equation below correct.

$$y_l = \mathrm{sgn}(\omega.x + b),\ i = 1,\ldots,l \quad (3)$$

The optimal hyperlane should make the distance between the closest vectors and the hyperplane maximal. To all training samples x_i , the minimum of $|\omega + b|$ is 1. So the minimal distance between the samples and the hyperplane is

$$|\omega.x + b| / ||\omega|| = 1 / ||\omega|| \quad (4)$$

The hyperplane should be constrained to

$$y_l [(\omega.x + b)] \geq 1,\ i = 1,...,l \quad (5)$$

The optimization condition of ω and b is to make the sum of the minimal distance between the two samples and the hyperplane $2/||\omega||$ maximal. Consider the situation of liner inseparable samples; we need to introduce slack variables $\xi \geq 0,\ i = 1,...,l$ to allow the margin constraints to be violated. So the constraint softens to

$$y_l [(\omega.x + b)] \geq 1 - \xi_i,\ i = 1,...,l \quad (6)$$

And the primal problem can be described as optimal problem [7, 14]:

$$\min_{\omega,b,\xi} \frac{1}{2}||\omega||^2 + C \cdot \sum_{i=1}^{l} \xi_i \quad (7)$$

$$s.t \quad y_i [(\omega.x + b)] \geq 1,\ i = 1,...,l \quad (8)$$

$$\xi_i \geq 0,\ i = 1,...,l$$

$C > 0$ is a punished parameter.

Using Lanrange multiplier method, the primal problem could be changed to its dual form:

$$\min_{\omega,b,\xi} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j a_i a_j (x_i x_j) - \sum_{j=1}^{l} a_j \quad (9)$$

$$\text{s.t} \quad \sum_{i=1}^{l} y_i a_i = 0 \quad (10)$$

$$0 \leq a_i \leq C,\ i = 1,...,l$$

Solve this optimal problem and acquire the optimal solution $a^* = (a_1^*,\ldots,a_l^*)$; compute

$$\omega^* = \sum_{i=1}^{l} y_i a_j^* x_i \quad (11)$$

Choose a positive a_j^* $(0 < a_j^* < C)$ and compute

$$b^* = y_j - \sum_{i=1}^{l} y_i a_j^* (x_i \cdot x_j) \quad (12)$$

The final decision function is

$$f(x) = \mathrm{sgn}(\omega^* \cdot x + b^*) \quad (13)$$

We will use The RBF kernel for our experiments:

$$k(x, x') = \exp(-\gamma||x - x'||^2), \quad (14)$$

With the RBF kernel (14), there are two parameters to be determined in the SVM model:

C and γ. To get good generalization ability, we conduct a validation process to decide parameters. The procedure is as the following:

1. Consider a grid space of (C, γ) with $\log_2 C \in \{-5,-3, \ldots, 15\}$ and $\log_2 \gamma \in \{-15,-13, \ldots, 3\}$.
2. For each hyperparameter pair (C, γ) in the search space, conduct 5-fold cross validation on the training set.
3. Choose the parameter (C, γ) that leads to the lowest CV balanced error rate.
4. Use the best parameter to create a model as the predictor.

5. Proposed System

In this section, we describe our proposed work, and also describe how to apply these methods to build detection patterns with the high performance for intrusion detection.

5.1. Overview of the framework

The proposed framework applies data mining techniques to build patterns for network intrusion detection. The system uses Random Forest (RF) for feature selection and parameter optimization and Support Vector Machine (SVM) for intrusion detection. An overall flow of proposed approach is shown in Figure 1.

Data preprocessing is utilized to do data arrangement. The preprocessed network audit data is divided into two dataset; training set and testing set. The training set is further separated into learning set and validation set. Although RF is not to perform cross-validation to get a balanced estimate of generalization error since RF is robust against over-fitting [6], the system will adopt n-fold cross validation to minimize that. The learning set is used to build intrusion detection models based on RF for feature selection. The RF produces the variable importance in numerical form. Feature ranking is performed according to the important value of features. So we can easily figure out which features are important or not. Then, the learning set with selected features is used to build intrusion detection models based on SVM. The testing set is used to evaluate the built detection models in terms of detection rates.

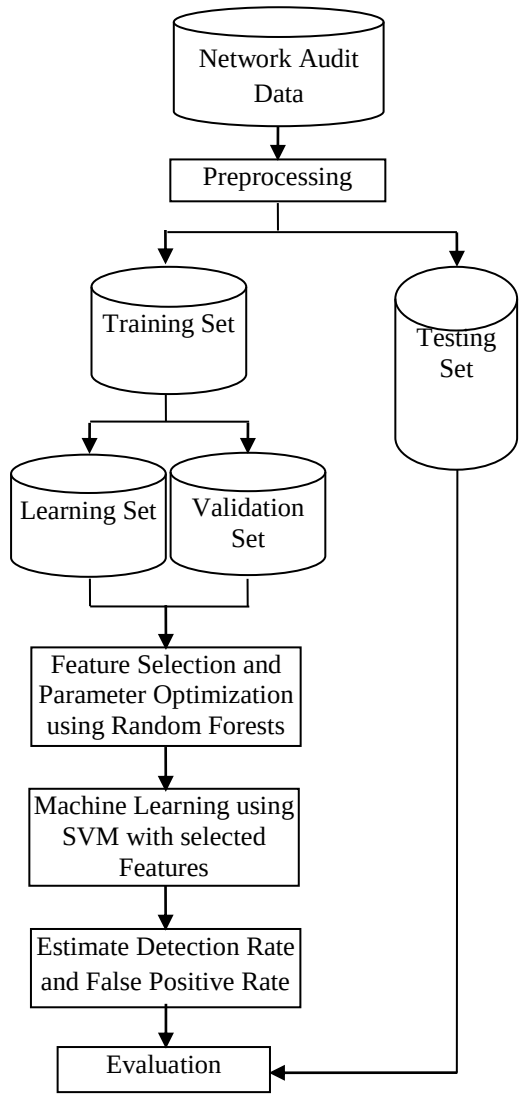

Figure 1. An overall flow of the proposed system

6. Intrusion Detection Dataset

6.1. KDD cup'99 Dataset

In this section, the performance of our approach will examined with KDD Cup 1999 intrusion detection dataset[5][14], which is most commonly used for evaluation. The KDD Cup 1999 Data is original from 1998 DARPA Intrusion Detection Evaluation. A process can be composed of many system calls. A system call is a text record. In this phase, some useless data will be filtered and modified. For example, some text items need to be converted into numbers. Every process in the database has 41 attributes and one class label. The labels of the 41 features and their corresponding network data features are shown in Table 1. The data set contains 24 attack types, which are categorized into four types as follows:

1. **Denial Of Service (DOS)**: In this type of attack legitimate user is denied to access a machine by making some computing resources or memory full. For example TCP SYN, Back etc.
2. **Remote to User (R2L)**: In this type of attack remote user tries to gain local access as the user of the machine. Foe example FTP_write, Guest etc.

3. **User to Root (U2R)**: In this type of attack the attacker tries to gain root access to the system. For example Eject, Fdformat etc.

4. **Probing**: In this type of attack attacker tries to scan a network of computer to fine known vulnerabilities or to gather information. For example Ipsweep, Mscan.

Table 1. Features of KDD cup'99 dataset

No	Feature Name	Type
1	duration	continuous
2	protocol_type	discrete
3	service	discrete
4	flag	discrete
5	src_bytes	continuous
6	dst_bytes	continuous
7	land	discrete
8	wrong_fragment	continuous
9	urgent	continuous
10	hot	continuous
11	num_failed_logins	continuous
12	logged_in	discrete
13	num_compromised	continuous
14	root_shell	discrete
15	su_attempted	discrete
16	num_root	continuous
17	num_file_creations	continuous
18	num_shells	continuous
19	num_access_files	continuous
20	num_outbound_cmds	continuous
21	is_host_login	discrete
22	is_guest_login	discrete
23	count	continuous
24	srv_count	continuous
25	serror_rate	continuous
26	srv_serror_rate	continuous
27	rerror_rate	continuous
28	srv_rerror_rate	continuous
29	same_srv_rate	continuous
30	diff_srv_rate	continuous
31	srv_diff_host_rate	continuous
32	dst_host_count	continuous
33	dst_host_srv_count	continuous
34	dst_host_same_srv_rate	continuous
35	dst_host_diff_srv_rate	continuous
36	dst_host_same_src_port_rate	continuous
37	dst_host_srv_diff_host_rate	continuous

38	dst_host_serror_rate	continuous
39	dst_host_srv_serror_rate	continuous
40	dst_host_rerror-rate	continuous
41	dst_host_srv_rerror_rate	continuous

6.2 Performance Measure

The system trains old data for new attack behaviors. The main estimating methods are precision and recall. The Network Intrusion Detection Systems estimates parameters shown in Table 2.

Table 2. The Network IDS Estimation

Parameter	Definition
True Positive Rate (TP)	Attack occur and alarm raised
False Positive Rate (FP)	No attack but alarm raised
True Negative Rate (TN)	No attack and no alarm
False Negative Rate (FN)	Attack occur but no alarm

To estimate the performance of the system, three important formulas are used to evaluate system accuracy [11]; attack detection rate (ADR), false positive rate (FPR) and system accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (15)$$

$$\text{ADR} = \frac{TP}{TP+FN} \times 100\% \quad (16)$$

$$\text{FPR} = \frac{FP}{FP+TN} \times 100\% \quad (17)$$

The experiments will test attack detection rate, false positive rate and accuracy among 41 features and reduced features.

7. Conclusion and Future Work

In this paper, we propose a new framework of network intrusion detection system and perform system performance on the KDD cup 99 intrusion detection dataset. We use Random Forests (RF) for feature selection and parameter optimization and Support Vector Machine (SVM) for intrusion detection. There is no approach which uses both Random Forests and Support Vector Machine for IDS.

In the future the proposed system will be implemented. The final results for the proposed system are not available, but the result of intrusion detection system could be faster than other methods.

References

[1] A. Zainal, M.A. Maarol and S. M. Shamsuddin, "Ensemble Classifiers for Network Intrusion Detection System", *Journal of Information Assurance and Security*, 2009.

[2] B. Lariviere, and D. Van den Poel, "Predicting Customer Retention and Profitability by Using Random Forests and Regression Forests Techniques." *Journal of Expert Systems with Applications*, Vol. 29, Issue 2, (August 2005) pp. 472-482.

[3] J. Peters, B. De Baets, N.E.C Verhoest, R. Samson, S. Degroeve, P. De Becker and W. Huybrechts, "Random Forests as a Tool for Ecohydrological Distribution Modelling." *Journal of Ecological Modelling*, Vol 207, Issue 2-4, October 2007, pp. 304-318.

[4] J. Zhang, and M. Zulkernine, 2006. A Hybrid Network Intrusion Detection Technique Using Random Forests. In Proceedings of the IEEE First International Conference on Availability, Reliability and Security (ARES'06).

[5] KDD Cup 1999 Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[6] L. Breiman, "Random Forests", *Machine Learning* 45(1):5–32, 2001.

[7] Nello Cristianini, John Shawe-Taylor, "An Introduction to Support Vector Machines and Other

Kernel–based Learning Methods", China Machine Press, 2005.
[8] P. Xu, and F. Jelinek, "Random Forests and the Data Sparseness Problem in Language Modeling." *Journal of Computer Speech and Language*, Vol. 21, Issue 1(Jan 2007) pp. 105-152.
[9] R. Agarwal and M.V. Joshi, "PNrule: a new framework for learning classifier models in data mining (a case-study in network intrusion detection)," *Proceedings of First SIAM Conference on Data Mining*, 2001.
[10] R.Chen, K. Cheng, and C. Hsieh, "Using Rough Set and Support Vector Machine for Network Intrusion Detection", *International Journal of Network Security & Its Applications (IJNSA)*, Vol 1, No 1, April 2009.
[11] R. C. Chen and S. P. Chen, "Intrusion Detection Using a Hybrid Support Vector Machine Based on Entropy and TF-IDF". *International Journal of Innovative Computing, Information and Control (IJICIC)*, Vol. 4, Number 2, pp. 413-424, 2008.
[12] S. A. Mulay, P.R. Devale, and G. V. Garje,"Intrusion Detection System Using Support Vector Machine and Decision Tree", *International Journal of Computer Applications(0975_8887)*, Volume 3_No.3, June 2010.
[13] S. Lee, D. Kim, and J. Park, "A Hybrid Approach for Real-Time Network Intrusion Detection Systems", *International Conference on Computational Intelligence and Security*, 2007.
[14] Steve R. Gunn, "Support Vector Machines for Classification and Regression", University of Southampton, 1998.
[15] YMahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of KDD CUP'99 data set. IEEE-2009.